Vincentius Charles Maynad
07111940000038
SPC

TUGAS NEURAL NETWORK

PRBS (Psuedorandom Binary Sequences)

PERSOALAN

Membuat sistem dengan plant s^2 + b^2s + c yang diberi input PRBS.

B dan C didapat dari NRP yang udah dibahas di kelas

Setelah dapat input, di buat matrix PRBS dan output, lalu proses NN (sama seperti tugas sebelumnya, tapi skrg ada 20 titik yang harus diiterasi sendiri-sendiri) --> goalnya dapet parameter di titik terakhir (20 itu)

epsilon katakan 10^-3. Tapi bebas..

Sebelum buat PRBS, uji open loop dulu (step, ramp, sin)

Harus ada 1 Ts >= ts --> contoh tadi berarti Ts >= 4 detik dan ts >= 0,4

Simulasi pakai simulink, tapi NN pakai matlab

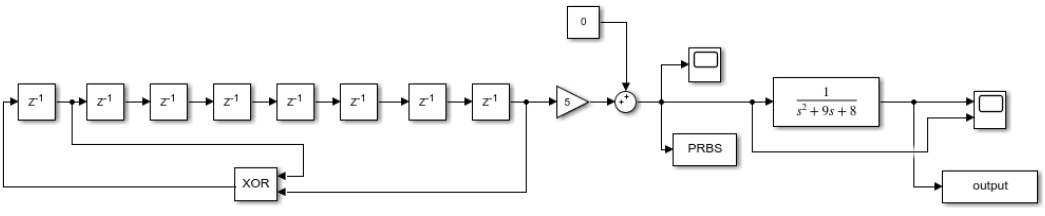
JAWABAN

I. Pembuatan sistem

Pada tugas kali ini, saya diminta untuk membuat sistem yang memiliki input sinyal biner berupa PRBS (Psuedorandom Binary Sequences). Plant yang saya gunakan didapat dari nomor NRP saya yang dituliskan sebagai :

$$H(s)=\frac{1}{s^2+9s+8}$$

Plant akan dihubungkan dengan PRBS – signal generator yang terbentuk dari susunan delay dan gain. Outputnya akan kita simpan ke MATLAB untuk diolah pada tahap selanjutnya. Saya merancang sistem saya menggunakan SIMULINK, berikut penampakannya :



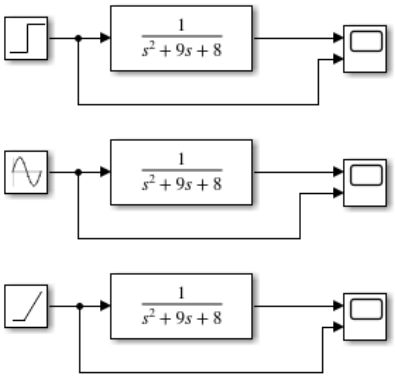
Gambar 1. Sistem orde dua dengan input PRBS

Adapun fungsi dari gain adalah memperkuat sinyal input karena sinyal aslinya bernilai cukup kecil. Saya pilih gain 5. Sedangkan konstanta 0 pada sebelah kanannya berfungsi hampir

serupa dengan gain namun disini hanya bersifat sebagai tambahan untuk kepentingan *plotting.* Bila kita berikan nilai, maka input akan menjadi PRBS + konstanta. Namun penambahan ini tidak akan merubah bentuk input yang dihasilkan setelah gain.

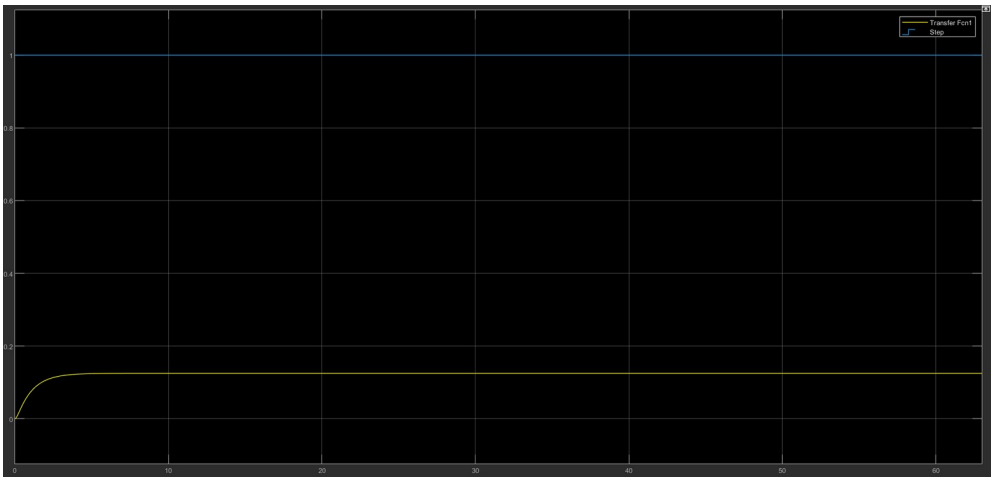
II. Pengujian pada sistem

Sebelum melangkah ke simulasi utama, ada baiknya kita melakukan pengujian terlebih dahulu terhadap plant yang dimiliki. Pengujian yang dilakukan adalah dengan memberikan input sinyal step, sinyal ramp, dan sinusoidal. Diharapkan kita dapat melihat respon dari sistem untuk menentukan penanganan yang diberikan. Plot dilakukan dengan waktu 63 detik.



Gambar 2. Rangkaian untuk pengujian

a. Sinyal step (u)

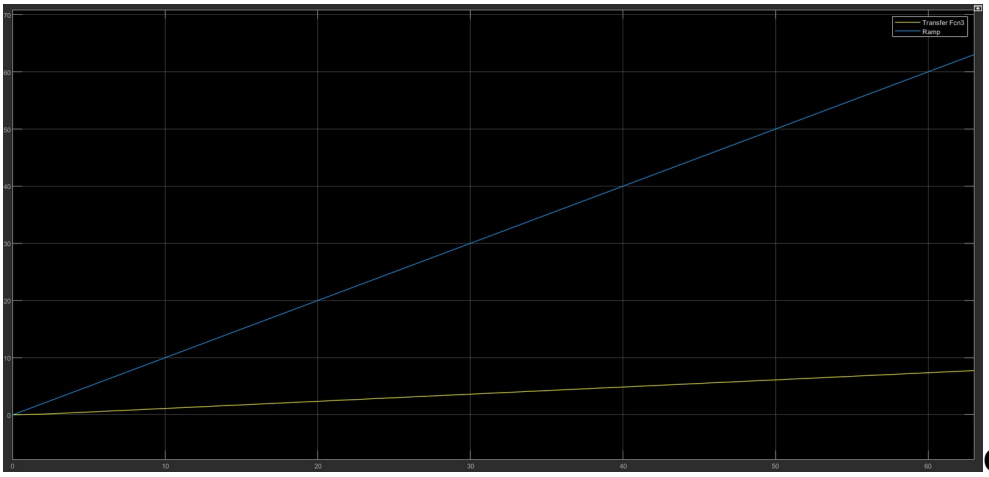


Gambar 3. Output uji step sistem

Sinyal input step adalah yang berwarna biru dan sinyal output plant adalah yang berwarna kuning. Dari plot tersebut, bisa langsung kita lihat bahwa output bernilai sangat kecil bila dibandingkan dengan *set – point* yang bernilai 1. Lalu, dengan bantuan program, kita juga tahu bahwa nilai maksimum

respon adalah sekitar 0,12 dengan *rise time* 2.184 sekon dan memiliki *overshot* yang sangat kecil, yakni 0.5%

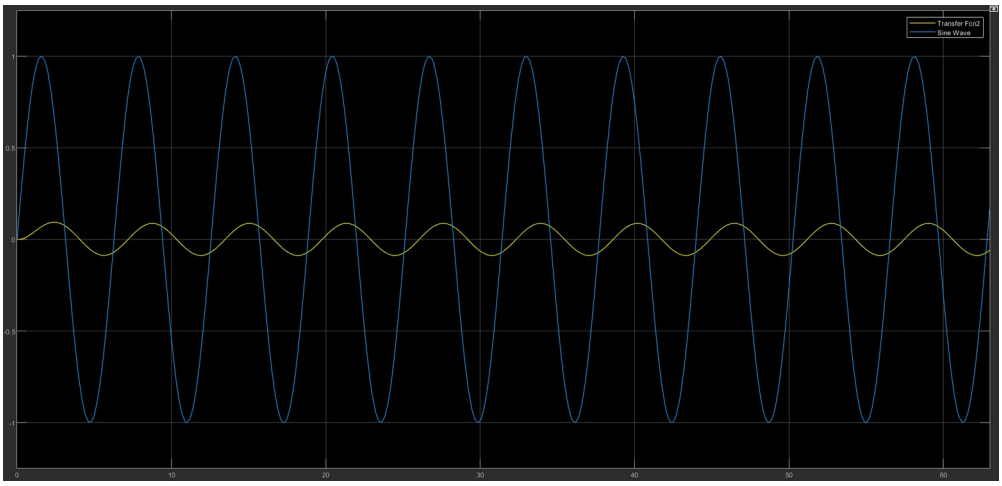
b. Sinyal ramp (r)



Gambar 4. Output uji ramp sistem

Masih dengan warna yang sama, berdasarkan output uji ramp, bisa kita lihat bahwa sinyal output sistem bernilai kecil sehingga tidak bisa mengejar sinyal input. Hal ini sesuai dengan output step.

c. Sinyal sinusoidal (sin)



Gambar 5. Output uji sinus sistem

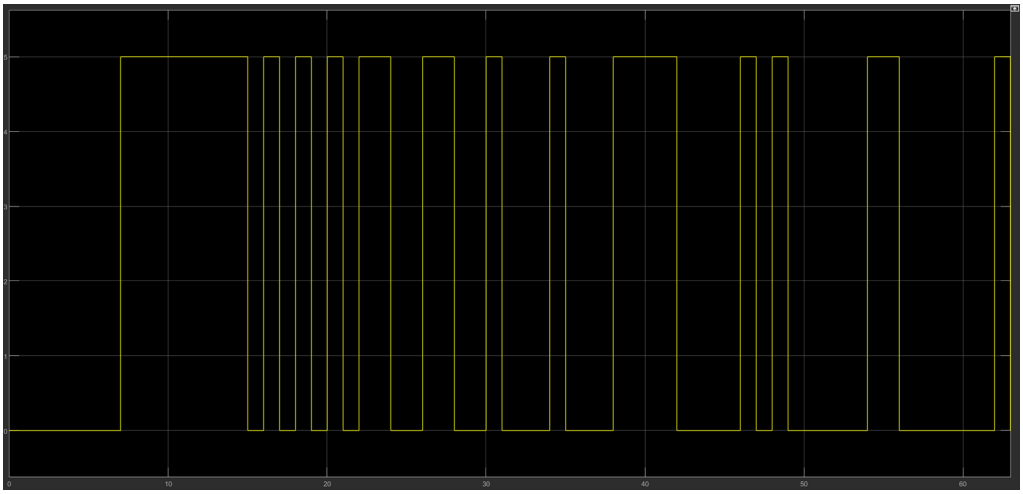
Terakhir adalah pengujian menggunakan sinyal sinusoidal dan menghasilkan output seperti gambar 5. Sama seperti sebelumnya, nilai dari output jauh lebih kecil daripada input. Selain itu, sepertinya terjadi sedikit *delay* sehingga output bergeser beberapa fasa, terlambat daripada input.

d. Analisa hasil

Berdasarkan ketiga pengujian, bisa disimpulkan kalau respon dari plant menghasilkan output yang bernilai kecil, selain itu terdapat sedikit *delay* yang membuat respon agak terlambat dibanding input walaupun tidak besar. Hal ini berarti kita perlu memberikan sinyal masukan yang lebih besar agar proses simulasi yang dilakukan lebih nyaman secara numerik. Caranya bisa dengan memberikan gain kepada input seperti yang kita lakukan pada sistem, yaitu memberi gain bernilai 5.

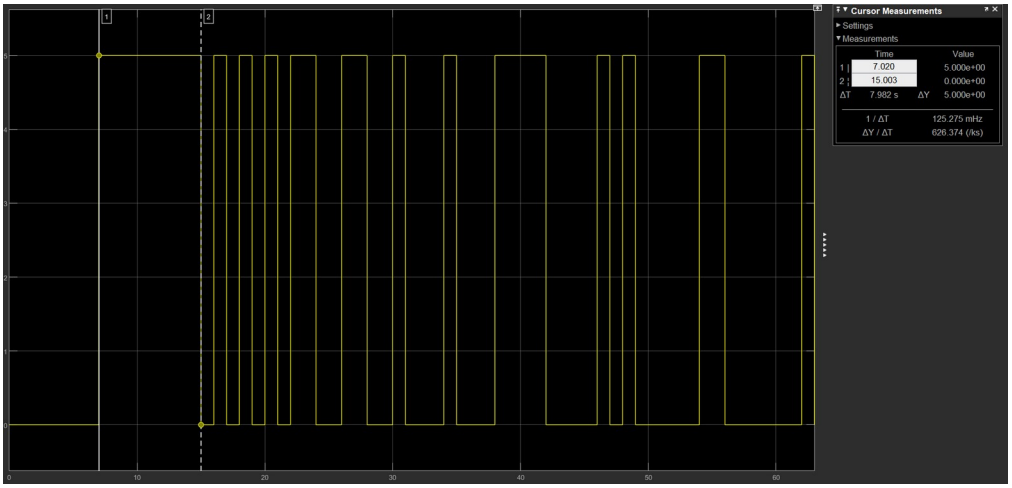
III. Pembangkitan sinyal biner (PRBS)

Sekarang, mari kita bangkitkan sinyal input yang akan digunakan pada tugas kali ini. PRBS adalah urutan biner yang (dihasilkan dengan algoritma deterministik) sulit diprediksi dan menunjukkan perilaku statistik yang mirip dengan urutan yang benar-benar acak. Menggunakan rangkaian pada gambar 1, kita plot sinyal PRBS kita.



Gambar 6. Sinyal biner yang akan digunakan sebagai input

Sinyal diplot sampai dengan 63 sekon. Terlihat bahwa sinyal adalah sinyal biner 1 dan 0 yang nilai amplitudenya berada diantara 0 dan 5. Hal itu disebabkan akibat penambahan gain.

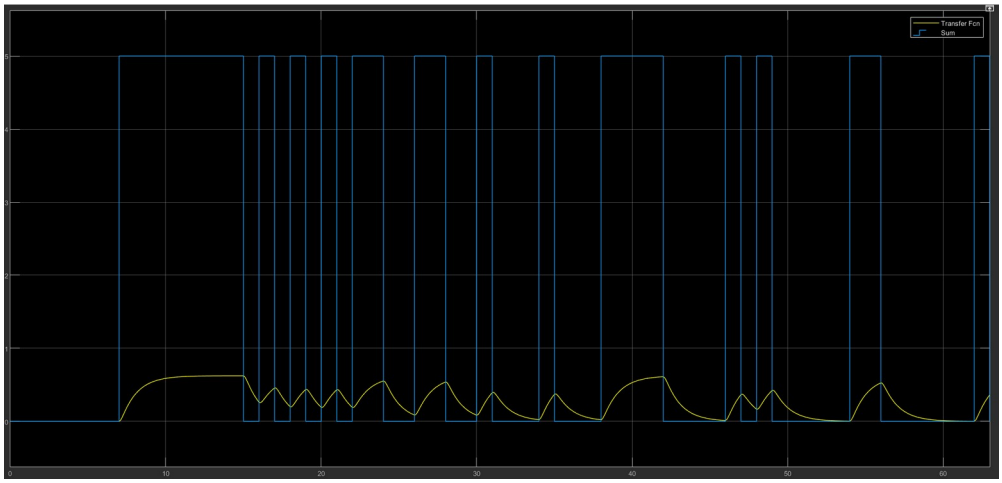


Gambar 7. Pembuktian $T_s \geq t_s$

Selain itu, sinyal juga sudah memenuhi minimal 1 $T_s \geq t_s$. Ts adalah waktu ketika sinyal berada pada nilai 1 (atas), sedangkan ts adalah waktu ketika sinyal berada pada nilai 0 (bawah). Pada gambar 7, diperlihatkan bahwa garis waktu pertama menunjukan waktu 7.02 sekon sedangkan garis waktu kedua menunjukan waktu 15.003 sekon. Dari situ bis akita hitung nilai Ts adalah 15.003 – 7.02 = 7,983 sekon. 7.02 sekon adalah ts dan dari plot, tidak ada ts yang memiliki waktu lebih lama. Demikian, input sudah siap diolah.

IV. Output sistem

Dengan input sudah ditentukan maka kita bisa langsung masukan ke plant sehingga menghasilkan output sebagai berikut.



Gambar 8. Output dari sistem dengan input PRBS

Output plant adalah yang berwarna kuning, sedangkan input PRBS berwarna biru. Bisa dilihat, output akan berdinamika naik dan turun

mengikuti bentuk dari input. Ketika bernilai 1, maka output akan naik hingga nilai saturasi tertentu, kemudian akan turun bila output menjadi 0.

V. Pengambilan data untuk *Neural Network*

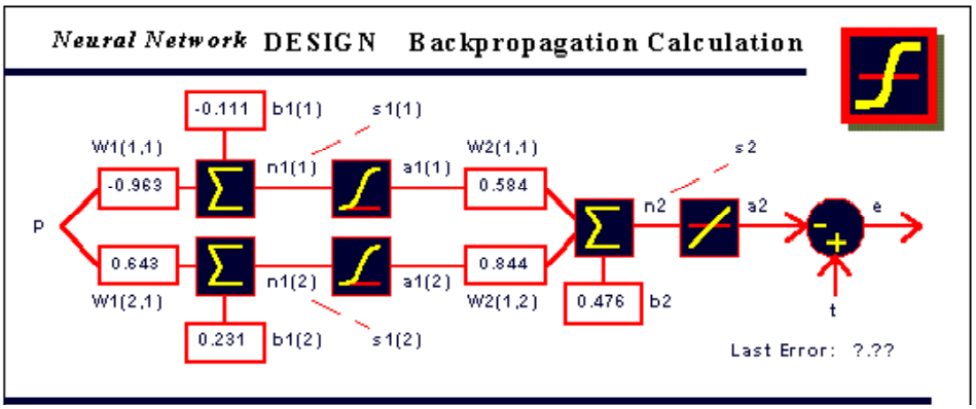
Output sudah didapatkan, nilai dari output akan digunakan sebagai salah satu parameter neural network, yaitu input neural network. Begitu pula input sistem akan kita gunakan sebagai target dari neural network. Pada tugas ini, akan dilakukan 20 periode *neural network* yang diambil dari 20 data input output sistem pertama. Datanya adalah sebagai berikut :

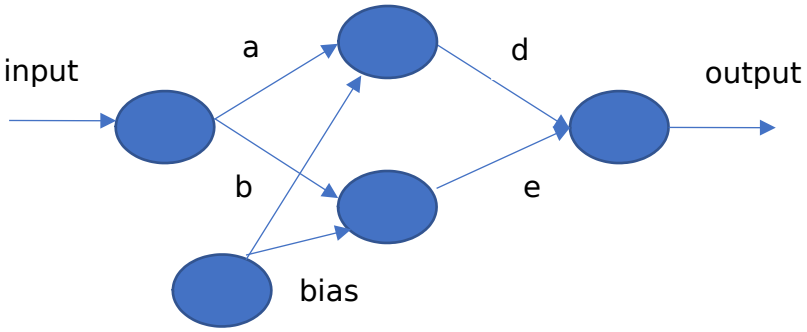
Tabel 1. Data input PRBS dan outputnya

Data ke -	PRBS	Output
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0.3263
9	5	0.5283
10	5	0.5894
11	5	0.6119
12	5	0.6202
13	5	0.6232
14	5	0.6243
15	5	0.6248
16	5	0.2627
17	0	0.4589
18	5	0.2016
19	0	0.4364
20	5	0.1934

VI. Komputasi *Neural Network*

Berdasarkan data pada tabel 1, kita bisa masukan ke sistem neural network sebagai berikut :





Gambar 9. Contoh ilustrasi sistem *neural network*

Berdasarkan ilustrasi, kita memerlukan beberapa parameter, yakni input, a, b, d, e, bias (3 bias). Selain itu, kita juga memerlukan nilai target sebagai *set – point* yang ingin dicapai dan *nilai error* yang ditoleransi. Sebagai inisiasi, kita coba samakan target dan input sesuai dengan data ke – 1 pada tabel 1. Sedangkan untuk nilai a, b, d, e dan bias kita dapatkan secara random menggunakan program MATLAB. Selain itu, kita pilih nilai faktor lupa sebesar α=0.1

a. Nilai parameter inisiasi

Dipilih nilai parameternya sebagai berikut :

#Target
T = Target = 0
#Synaptic weights
a = W11 = 0.9572
b = W12 = 0.4854
d = W21 = 0.8003
e = W22 = 0.1419
#bias
$b1_1$ = bias = 0.4218
$b1_2$ = bias = 0.9157
b2 = bias = 0.7922
#input
p = input = 0
#error yang dipilih sebesar 0.001 atau 10^{-3}
α=0.1

Vincentius Charles Maynad
07111940000038
SPC

b. Percobaan dan iterasi

Tabel 2. Update parameter 20 periode *neural network*

Periode ke -	Parameter						
	Update Parameter						
	W11	W12	W21	W22	$b1_1$	$b1_2$	b2
1 (32)	0.9572	0.4854	0.8003	0.1419	0.1634	0.8771	-0.5322
2 (1)	0.9572	0.4854	0.8003	0.1419	0.1634	0.8771	-0.5322
3 (1)	0.9572	0.4854	0.8003	0.1419	0.1634	0.8771	-0.5322
4 (1)	0.9572	0.4854	0.8003	0.1419	0.1634	0.8771	-0.5322
5 (1)	0.9572	0.4854	0.8003	0.1419	0.1634	0.8771	-0.5322
6 (1)	0.9572	0.4854	0.8003	0.1419	0.1634	0.8771	-0.5322
7 (1)	0.9572	0.4854	0.8003	0.1419	0.1634	0.8771	-0.5322
8 (13)	0.9545	0.4850	0.7859	0.1275	0.1551	0.8759	-0.5764
9 (18)	1.2991	0.6351	2.1556	1.4972	0.8075	1.1599	2.0162
10 (8)	1.2964	0.6331	2.1467	1.4883	0.8028	1.1566	2.001
11 (6)	1.2954	0.6324	2.1433	1.4849	0.8012	1.1554	1.9956
12 (4)	1.295	0.6321	2.1421	1.4837	0.8006	1.1550	1.9937
13 (3)	1.2949	0.6320	2.1417	1.4833	0.8004	1.1548	1.9929
14 (1)	1.2949	0.6320	2.1416	1.4832	0.8003	1.1548	1.9927
15 (1)	1.2948	0.6320	2.1415	1.4831	0.8003	1.1548	1.9926
16 (15)	1.3082	0.6405	2.1757	1.5173	0.8513	1.1870	2.1230
17 (22)	0.9114	0.4216	0.8845	0.2261	-0.0135	0.7102	-0.6908
18 (27)	1.1090	0.5010	1.6292	0.9708	0.9669	1.1037	3.0032
19 (23)	0.8572	0.4003	0.3279	-0.3305	0.39	0.8730	0.0214
20 (29)	0.9598	0.3987	1.0762	0.4178	0.9202	0.8645	3.8902

Tabel diatas adalah hasil 20 kali percobaan neural network dengan input dan output yang berbeda. Adapun angka didalam kurung () disebelah periode adalah jumlah iterasi yang dilakukan pada periode tersebut. Saya hanya

menyertakan data update parameter saja karena keterbatasan tempat dan saya kira itulah data yang kita perlukan untuk diamati.

VII. Hasil akhir dan analisa

Berdasarkan hasil tabel 2, kita dapatkan nilai parameter akhirnya setelah 20 kali percobaan adalah sebagai berikut :

#Synaptic weights
a = W11 = 0.9598
b = W12 = 0.3987
d = W21 = 1.0762
e = W22 = 0.4178
#bias
$b1_1$ = bias = 0.9202
$b1_2$ = bias = 0.8645
b2 = bias = 3.8902

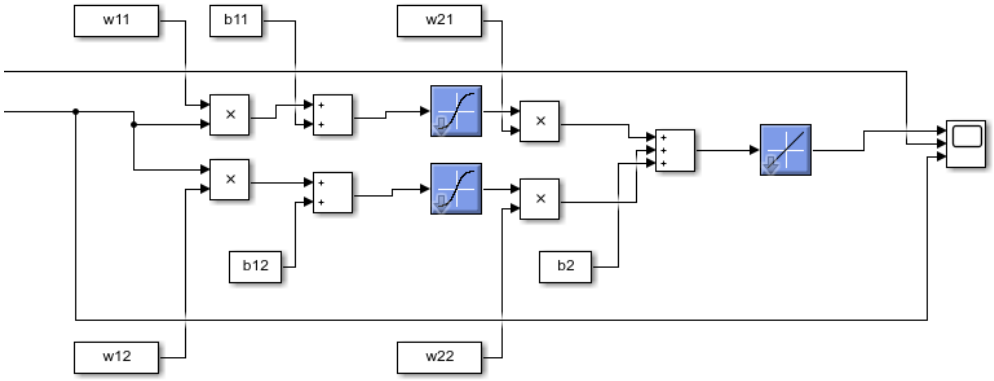
Parameter diatas menunjukan nilai peran dari tiap parameter terhadap data yang diolah. Kita ambil contoh W21 yang memiliki nilai bahkan lebih dari 1, berarti perannya sangat penting dalam perhitungan, sebaliknya W12 yang hanya bernilai 0.3987 menandakan kurangnya kepentingan nilainya. Sedangkan pada bias, didapatkan 3 nilai yang cukup tinggi, bahkan b2 bernilai 3. Hal ini berarti terjadi cukup banyak pembulatan dalam komputasi sehingga diperlukan nilai bias yang tinggi agar perhitungan tetap terjaga keakuratannya dari pengaruh pembulatan.

Sekarang, kita lihat tabel 2 kembali, pada periode ke – 1, terjadi perubahan nilai parameter karena nilai bias, walaupun target dan input sama – sama bernilai 0. Lalu pada periode ke – 2 sampai 7 tidak ada perubahan karena nilai input dan target yang masih sama. Barulah pada periode ke – 8 terjadi perubahan nilai karena adanya perubahan input. Kemudian terjadi perubahan lagi karena nilai target berubah menjadi 5 pada periode ke – 9. Nilai parameter tidak banyak berubah dari periode ke – 10 sampai 16. Meskipun nilai input berubah, nilai target masih sama sehingga perubahan hanya kecil. Hal yang menarik adalah jumlah iterasinya semakin menurun seiring berjalannya periode. Ini berarti sistem sudah mulai terbiasa dengan keadaan, kecuali pada periode ke – 16 karena merupakan masa transisi dari 5 ke 0. Pada periode – periode akhir, nilai target berubah – ubah, hal ini membuat sistem harus bekerja ekstra untuk menyesuaikan diri. Ditunjukan dengan jumlah iterasi yang cukup banyak dan perubahan parameter yang naik dan turun.

Bila kita bandingkan parameter akhir dengan parameter awal, ada beberapa yang berubah secara signifikan seperti d, e, b11, dan b2. Namun juga ada yang tidak, seperti a,b, dan b12.

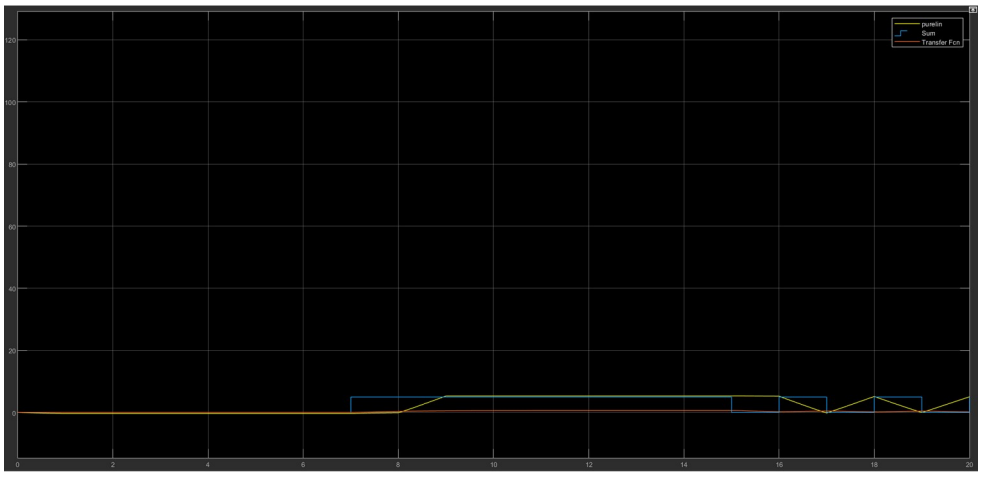
VIII. Simulasi parameter *Neural Network* pada sistem

Setelah mendapat nilai parameter baru, kita coba implementasikan model neural network pada SIMULINK. Sesuai ilustrasi gambar 9, kita bentuk sistem sebagai berikut :

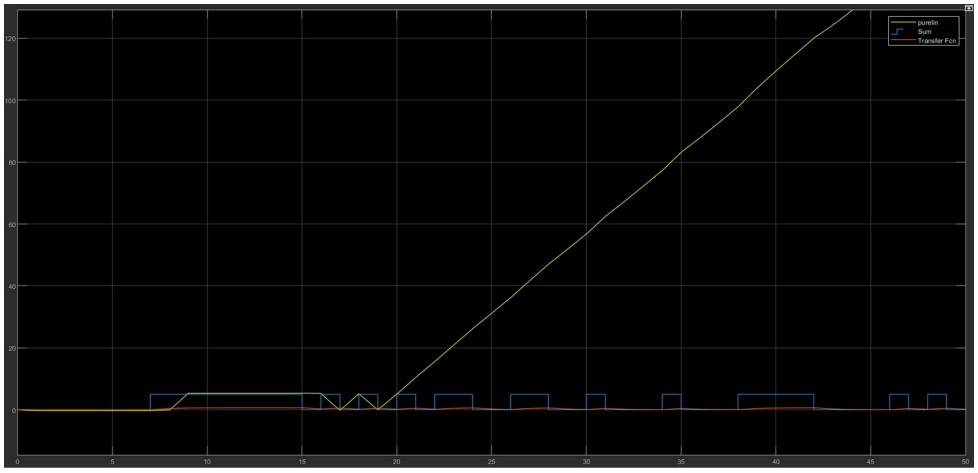


Gambar 10. Model *neural network*

Input yang diberikan adalah output dari plant, dialirkan ke dua cabang W. Scope akan membandingkan sinyal input, sinyal PRBS, dan sinyal output yang baru.



Gambar 11. Input, PRBS asli, dan output *neural network* selama 20 detik



Gambar 12. Input, PRBS asli, dan output *neural network* selama 50 detik

Saya menggunakan dataset weight pada tiap variabel yang dituliskan di *workspace*. Adapun datanya hanya sampai detik ke – 20 sesuai dengan data pada tabel 2.

Dapat dilihat dari gambar 11, bahwa output yang dihasilkan oleh neural network sudah menyerupai sinyal input PRBS yang diberikan walaupun ada beberapa error. Hasilnya berupa sinyal segitiga karena kita menggunakan waktu dalam satuan diskrit dan weight serta bias hanya berlaku pada detik tersebut. Ketika PRBS bernilai 1, nilai output naik mendekati 5, ketika bernilai 0, nilai output turun mendekati 0. Mungkin data bisa lebih baik bila kita menggunakan rentang waktu yang lebih spesifik diantara periodenya.

Lalu kita lihat gambar 12, output sudah tidak lagi terkontrol begitu melewati 20. Hal ini disebabkan data weight yang digunakan hanya terbatas sampai 20 sekon sehingga bila lebih, tidak ada data yang digunakan untuk menkontrol sinyal tersebut. Hal ini bisa diatasi dengan menggunakan lebih banyak data weight.

IX. Kesimpulan

Metode neural network adalah metode yang terus belajar untuk menyesuaikan diri dengan kondisi. Metode ini akan langsung merubah parameter dengan signifikan bila kondisi berubah drastis. Demikian pula bila kondisi tidak berbeda jauh, maka jumlah perhitungan untuk kondisi berikutnya akan selalu lebih cepat dibanding sebelumnya. Nilai parameter yang tinggi menunjukan tingkat kepentingannya pada sistem, begitu pula sebaliknya. Output dari sistem bergantung kepada nilai parameter tersebut.

Vincentius Charles Maynad
07111940000038
SPC

LAMPIRAN

Berikut kode program MATLAB yang saya gunakan untuk perhitungan neural network pada periode ke – 1.

```
% initial parameter
%weighted
w11 = 0.9572;
w12 = 0.4854;
w21 = 0.8003;
w22 = 0.1419;
%bias
b11 = 0.4218;
b12 = 0.9157;
b2  = 0.7922;
%init error and input
error = 1;
p = 0;
%target output
Target = 0;
iter = 0;
while 1==1
    iter = iter + 1
    if abs(error) <= 0.001 %pemilihan error output
        break
    else
        %forward
        if iter == 1
            N11 = w11*p+b11
            A11 = logsig(N11)
            N12 = w12*p+b12
            A12 = logsig(N12)
            A2 = purelin((w21*A11+w22*A12)+b2)%A2 =
purelin((d*A11+e*A12+f*A13)+b1)
            error = Target-A2
            %Backpropagation
            S2 =-2*1*error
            S11 = (1-A11)*A11*(w21*S2)
            S12 = (1-A12)*A12*(w22*S2)
            %update param
            alpha = 0.1;
            w11 = w11-alpha*S11*p
            b11 = b11-alpha*S11
            w12 = w12-alpha*S12*p
            b12 = b12-alpha*S12
            w21 = w21-alpha*S2*p
            b2 = b2-alpha*S2
            w22 = w22-alpha*S2*p
        else
            N11 = w11*p+b11
            A11 = logsig(N11)
            N12 = w12*p+b12
```

Vincentius Charles Maynad
07111940000038
SPC

```
            A12 = logsig(N12)
            A2 = purelin((w21*A11+w22*A12)+b2) %A2 =
purelin((d*A11+e*A12+f*A13)+b2)
            error = Target-A2
            %Backpropagation
            S2 =-2*1*error
            S11 = (1-A11)*A11*(w21*S2)
            S12 = (1-A12)*A12*(w22*S2)
            %update param
            alpha = 0.1;
            w11 = w11-alpha*S11*p
            b11 = b11-alpha*S11
            w12 = w12-alpha*S12*p
            b12 = b12-alpha*S12
            w21 = w21-alpha*S2*p
            b2 = b2-alpha*S2
            w22 = w22-alpha*S2*p
        end
    end
end
```